This application is submitted in the name of Inventor Jerrold Von Hauck, Assignor to Apple Computer Inc., a California Corporation.

SPECIFICATION

GAP COUNT ANALYSIS FOR THE P13943a BUS

FIELD OF THE INVENTION

[0001] The present invention relates broadly to serial bus performance. Specifically, the present invention relates to improving bus performance by calculating the optimal gap_count parameter for a given topology utilizing a high-speed serial bus to connect devices.

BACKGROUND OF THE INVENTION

[0002] The Institute of Electrical and Electronic Engineers (IEEE) has promulgated a number of versions of a high-speed serial bus protocol falling under the IEEE 1394 family of standards (referred to herein collectively as "1394"). A typical serial bus having a1394 architecture interconnects multiple node devices via point-to-point links, such as cables, each connecting a single node on the serial bus to another node on the serial bus. Data packets are propagated throughout the serial bus using a number of point-to-point transactions, such that a node that receives a packet from another node via a first point-to-point link retransmits the received packet via other point-to-point links. A tree network configuration and associated packet handling protocol ensures that each node receives every packet once. The 1394-compliant serial bus may be used as an alternate bus for the parallel backplane of a computer system, as a low cost peripheral bus, or as a bus bridge between

architecturally compatible buses. Bus performance is gauged by throughput, or the amount of data that can be transmitted over the bus during a period of time.

[0003] There are several ways to improve bus performance. Devices connected to the bus can be arranged to minimize the longest round-trip delay between any two leaf nodes. This may involve either minimizing the number of cable connections between the farthest devices, reducing cable lengths, or both. Another way to improve bus performance is to group devices with identical speed capabilities next to one another. This avoids the creation of a "speed trap" when a slower device lies along the path between the two faster devices. Finally, bus performance can be improved by setting the PHY gap count parameter to the lowest workable value for a particular topology. However, determining this lowest workable value is problematic in that all of the variables affecting this value are unknown. Gap count parameters have been configured in the past using a subset of all possible variables, and the result is that the gap count is not optimal.

SUMMARY OF THE INVENTION

[0004] The present invention provides an optimal gap count that allows a high-speed serial bus to run faster and thus realize superior performance over prior buses. In an embodiment, bus management software sends a special PHY configuration packet that is recognized by all PHYs on the bus. The configuration packet contains a gap count value that all PHYs on the bus can use. As this gap count value decreases the time interval between packets that are transmitted, more real data can be transmitted over the bus per unit of time.

[0005] In an embodiment, the bus manager pings a PHY. The PHY sends a response to the ping, and a flight time value of the response from the PHY to the bus manager is added to calculate a round trip delay value. The ping command runs at the link layer level, from the link layer of one node to the link layer of another

node. All flight time between link layer and PHY is ignored, and just the flight time from one PHY to another PHY is calculated. The ping time measured shows the link-to-link delay. The delay between the bus and the link is specified in the bus standard with minimum and maximum values. The PHY and link layer of a node is designed to be within that range specified by the standard. The round trip delay between nodes can be calculated as:

$$Round\_Trip\_Delay_{\max}^{[P_X \mathrm{O} P_Y]} \leq \begin{pmatrix} Round\_Trip\_Delay_{Ping,\max}^{[P_{BM} \mathrm{O} P_X]} + \sum_{n}^{(BM,X)} 2 \cdot Jitter_n + \\ Round\_Trip\_Delay_{Ping,\max}^{[P_{BM} \mathrm{O} P_Y]} + \sum_{n}^{(BM,Y)} 2 \cdot Jitter_n + \\ PHY\_DELAY_{N,\max}^{P'_N \to P_N} + ARB\_RESPONSE\_DELAY_{N,\max}^{P_N \to P'_N} \\ \end{pmatrix} - \begin{pmatrix} 2 \cdot Round\_Trip\_Delay_{Ping,\min}^{[P_{BM} \mathrm{O} P_N]} + 4 \cdot Jitter_N + \\ PHY\_DELAY_{N,\min}^{P_N^{BM} \to P'_N} + ARB\_RESPONSE\_DELAY_{N,\min}^{P'_N \to P_N^{BM}} + \\ PHY\_DELAY_{N,\min}^{P_N^{BM} \to P_N} + ARB\_RESPONSE\_DELAY_{N,\min}^{P_N \to P_N^{BM}} \end{pmatrix}$$

[0006] This value can be communicated as the gap_count parameter contained in the configuration packet, thus setting the gap between packets to an optimal value and increasing bus performance.

[0007] Many other features and advantages of the present application will become apparent from the following detailed description considered in conjunction with the accompanying drawings, in which:

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] FIG. 1 illustrates an intervening path model between two nodes, X & Y, and denotes the reference points required for a full analysis;

[0009] FIG. 2 illustrates ack/iso gap preservation, in the case where PHY X originated the most recent packet and PHY Y is responding (either with an ack or the next isochronous arbitration/packet).

[0010] FIG. 3 illustrates the sequence PHY Y will follow in responding to a received packet.

[0011] FIG. 4 illustrates subaction gap preservation, in the case where PHY X originated the most recent packet and PHY Y is responding after a subaction gap with arbitration for the current fairness interval.

[0012] FIG. 5 illustrates consistent subaction gap detection, in the case where PHY X originates an isochronous packet, observes a subaction_gap, and begins to drive an arbitration indication.

[0013] FIG. 6 illustrates an internal gap detection sequence, by showing the timing reference for relating the external gap detection times to the internal gap detection times.

[0014] FIG. 7 illustrates consistent arbitration reset gap detection, in the case where PHY X originates an asynchronous packet, observes an arbitration reset gap, and begins to drive an arbitration indication.

[0015] FIG. 8 illustrates a ping subaction issued by the link in Node X and directed to Node Y.

[0016] FIG. 9 illustrates a Bus Manager Leaf to Leaf topology.

[0017] FIG. 10 illustrates a topology where the bus manager is not a leaf but is part of the connecting path between the two leaves.

[0018] FIG. 11 illustrates a topology where the bus manager is not a leaf but is not part of the connecting path between the two leaves.

DETAILED DESCRIPTION

[0019] Four well known limiting corner cases for gap count are examined in an effort to find the minimum allowable gap count for a given topology. Both the table method and pinging method of determining the optimal gap count are explored.

[0020] It is important to note that this analysis assumes that PHY_DELAY can never exceed the maximum published in the PHY register set. However, corner conditions have been identified in which it is theoretically possible to have PHY_DELAY temporarily exceed the maximum published delay when repeating minimally spaced packets. Although not a rigorous proof, this phenomena is ignored for this analysis on the basis that it is presumed to be statistically insignificant.

[0021] The path between any two given PHYs can be represented as a daisy chain connection of the two devices with zero or more intervening, or repeating, PHYs. FIG. 1 illustrates such a path between two nodes, X & Y, and denotes the reference points required for a full analysis.

Table 1: Variable Definitions

$ARB\_RESPONSE\_DELAY_n^{P_n \rightarrow P'_n}$	Delay in propagating arbitration indication received from port P_n of PHY n to port P'_n of PHY n.
$BASERATE_n$	Fundamental operating frequency of PHY n.
$cable\_delay_n$	One-way flight time of arbitration and data signals through $cable_n$. The flight-time is assumed to be constant from one transmission to the next and symmetric.
$DATA\_END\_TIME_n^{P_n}$	Length of DATA_END transmitted on port P_n of PHY n.
$PHY\_DELAY_n^{P'_n \rightarrow P_n}$	Time from receipt of first data bit at port P'_n of PHY n to re-transmission of same bit at port P_n of PHY n.
$RESPONSE\_TIME_n^{P'_n}$	Idle time at port P'_n of PHY n between the reception of a inbound packet and the associated outbound arbitration indication for the subsequent packet intended to occur within the same isochronous interval or asynchronous subaction.

[0022] For any given topology, the gap count must be set such that an iso or ack gap observed/generated at one PHY isn't falsely interpreted as a subaction gap by another PHY in the network. Ack/Iso gaps are known to be at their largest nearest the PHY that originated the last packet. To ensure that the most recent originating PHY doesn't interrupt a subaction or isochronous interval with asynchronous arbitration, its subaction_gap timeout must be greater than the largest IDLE which can legally occur within a subaction or isochronous interval. FIG. 2

illustrates the case in which PHY X originated the most recent packet and PHY Y is responding (either with an ack or the next isochronous arbitration/packet).

[0023] For all topologies, the idle time observed at point Px must not exceed the subaction gap detection time:

$$Idle_{\max}^{Px} < subaction\_gap_{\min}^{Px} \tag{1}$$

[0024] The idle time at point Px can be determined by examining the sequence of time events in the network. All timing events are referenced to the external bus (as opposed to some internal point in the PHY).

t_0 First bit of packet sent at point P_x

t_1 Last bit of packet sent at point P_x, DATA_END begins. t_1 follows t_0 by the length of the packet timed in PHY X's clock domain.

t_2 DATA_END concludes at point P_x, IDLE begins. t_2 follows t_1 by $DATA\_END\_TIME_X^{Px}$

t_3 First bit of packet received at point P'_Y. t_3 follows t_0 by all intervening cable_delay and PHY_DELAY instances.

t_4 Last bit of packet received at point P'_Y. t_4 follows t_3 by the length of the packet timed in PHY Y-1's clock domain.

t_5 DATA_END concludes at point P'_Y, gap begins. t_5 follows t_4 by $DATA\_END\_TIME_{Y-1}^{P_{Y-1}}$

t_6 PHY Y responds with ack packet, isoch packet, or isoch arbitration within $RESPONSE\_TIME_Y^{P'_Y}$ following t_5

t_7 Arbitration indication arrives at point P_x. t_7 follows t_6 by the all intervening cable_delay and ARB_RESPONSE_DELAY instances.

$$t_1 = t_0 + \frac{packet\_length}{packet\_speed \cdot BASERATE_X} \tag{2}$$

$$\begin{aligned} t_2 &= t_1 + DATA\_END\_TIME_X^{P_X} \\ &= t_0 + \frac{packet\_length}{packet\_speed \cdot BASERATE_X} + DATA\_END\_TIME_X^{P_X} \end{aligned} \tag{3}$$

$$t_3 = t_0 + cable\_delay_X + \sum_{n=X+1}^{Y-1} \left(cable\_delay_n + PHY\_DELAY_n^{P_n' \to P_n} \right) \tag{4}$$

$$\begin{aligned} t_4 &= t_3 + \frac{packet\_length}{packet\_speed \cdot BASERATE_{Y-1}} \\ &= t_0 + cable\_delay_X + \sum_{n=X+1}^{Y-1} \left(cable\_delay_n + PHY\_DELAY_n^{P_n' \to P_n} \right) + \\ &\quad \frac{packet\_length}{packet\_speed \cdot BASERATE_{Y-1}} \end{aligned} \tag{5}$$

$$\begin{aligned} t_5 &= t_4 + DATA\_END\_TIME_{Y-1}^{P_{Y-1}} \\ &= t_0 + cable\_delay_X + \sum_{n=X+1}^{Y-1} \left(cable\_delay_n + PHY\_DELAY_n^{P_n' \to P_n} \right) + \\ &\quad \frac{packet\_length}{packet\_speed \cdot BASERATE_{Y-1}} + DATA\_END\_TIME_{Y-1}^{P_{Y-1}} \end{aligned} \tag{6}$$

$$\begin{aligned} t_6 &= t_5 + RESPONSE\_TIME_Y^{P_Y'} \\ &= t_0 + cable\_delay_X + \sum_{n=X+1}^{Y-1} \left(cable\_delay_n + PHY\_DELAY_n^{P_n' \to P_n} \right) + \\ &\quad \frac{packet\_length}{packet\_speed \cdot BASERATE_{Y-1}} + DATA\_END\_TIME_{Y-1}^{P_{Y-1}} + RESPONSE\_TIME_Y^{P_Y'} \end{aligned} \tag{7}$$

$$t_7 = t_6 + \sum_{n=X+1}^{Y-1}\left(cable\_delay_n + ARB\_RESPONSE\_DELAY_n^{P_n \to P_n'}\right) + cable\_delay_X$$
$$= t_0 + \sum_{n=X+1}^{Y-1}\left(2 \cdot cable\_delay_n + PHY\_DELAY_n^{P_n' \to P_n} + ARB\_RESPONSE\_DELAY_n^{P_n \to P_n'}\right) +$$
$$2 \cdot cable\_delay_X + \frac{packet\_length}{packet\_speed \cdot BASERATE_{Y-1}} + DATA\_END\_TIME_{Y-1}^{P_{Y-1}} +$$
$$RESPONSE\_TIME_Y^{P_Y'} \quad (8)$$

Given t_0 through t_7 above, the Idle time seen at point P_x is given as:

$$Idle^{P_X} = t_7 - t_2$$
$$= \sum_{n=X+1}^{Y-1}\left(2 \cdot cable\_delay_n + PHY\_DELAY_n^{P_n' \to P_n} + ARB\_RESPONSE\_DELAY_n^{P_n \to P_n'}\right) +$$
$$2 \cdot cable\_delay_X + RESPONSE\_TIME_Y^{P_Y'} +$$
$$DATA\_END\_TIME_{Y-1}^{P_{Y-1}} - DATA\_END\_TIME_X^{P_X} +$$
$$\frac{packet\_length}{packet\_speed} \cdot \left(\frac{1}{BASERATE_{Y-1}} - \frac{1}{BASERATE_X}\right) \quad (9)$$

Let:

$$DE\_delta^{[P_{Y-1}, P_X]} = DATA\_END\_TIME_{Y-1}^{P_{Y-1}} - DATA\_END\_TIME_X^{P_X} \quad (10)$$

$$PPM\_delta^{[Y-1, X]} = \frac{packet\_length}{packet\_speed} \cdot \left(\frac{1}{BASERATE_{Y-1}} - \frac{1}{BASERATE_X}\right) \quad (11)$$

$$Round\_Trip\_Delay^{[P_X \, 0 \, P_Y]} = \sum_{n=X+1}^{Y-1}\left(\begin{array}{l} 2 \cdot cable\_delay_n + PHY\_DELAY_n^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_n^{P_n \to P_n'} \end{array}\right) + 2 \cdot cable\_delay_X \quad (12)$$

Then,

$$Idle^{P_X} = Round\_Trip\_Delay^{[P_X \circ P_Y]} + RESPONSE\_TIME_{Y}^{P_Y'} + DE\_delta^{[P_{Y-1}, P_X]} + PPM\_delta^{[Y-1, X]} \tag{13}$$

Substituting into Equation (1), Ack and Iso gaps are preserved network-wide if and only if:

$$\begin{bmatrix} Round\_Trip\_Delay^{[P_X \circ P_Y]} + RESPONSE\_TIME_{Y}^{P_Y'} + \\ DE\_delta^{[P_{Y-1}, P_X]} + PPM\_delta^{[Y-1, X]} \end{bmatrix}_{\max} < subaction\_gap_{\min}^{P_X} \tag{14}$$

[0025] The minimum subaction_gap at point P_x isn't well known. IEEE1394-1995, in Table 4-33, defines the minimum subaction_gap timeout used at a PHY's internal state machines, not at the external interface. It has been argued that the internal and external representations of time may differ by as much as ARB_RESPONSE_DELAY when a PHY is counting elapsed time between an internally generated event and an externally received event. However, the ARB_RESPONSE_DELAY value for a particular PHY isn't generally known externally. Fortunately, the ARB_RESPONSE_DELAY value for a PHY whose FIFO is known to be empty is bounded by the worst case PHY_DELAY reported within the PHY register map. This suggests a realistic bound for the minimum subaction_gap referenced at point P_x:

$$subaction\_gap_{\min}^{P_X} \geq subaction\_gap_{\min}^{i_X} - PHY\_DELAY_{X,\max}^{P_X} \tag{15}$$

where

$$subaction\_gap_{\min}^{i_X} = \frac{27 + gap\_count \cdot 16}{BASERATE_{X,\max}} \tag{16}$$

Combing Equations (14), (15), and (16):

$$\begin{bmatrix} Round\_Trip\_Delay^{[P_X \circ P_Y]} + \\ RESPONSE\_TIME_Y^{P_Y'} + \\ DE\_delta^{[P_{Y-1}, P_X]} + \\ PPM\_delta^{[Y-1, X]} \end{bmatrix}_{\max} < \left[\frac{27 + gap\_count \cdot 16}{BASERATE_{X,\max}} - PHY\_DELAY_{X,\max}^{P_X} \right] \tag{17}$$

Solving for gap_count:

$$gap\_count > \frac{BASERATE_{X,\max} \cdot \begin{bmatrix} Round\_Trip\_Delay_{\max}^{[P_X \circ P_Y]} + \\ \begin{bmatrix} RESPONSE\_TIME_Y^{P_Y'} + \\ DE\_delta^{[P_{Y-1}, P_X]} + PPM\_delta^{[Y-1, X]} \end{bmatrix}_{\max} + \\ PHY\_DELAY_{X,\max}^{P_X} \end{bmatrix} - 27}{16} \tag{18}$$

[0026] Since RESPONSE_TIME, DE_delta, and PPM_delta are not independent parameters, the maximum of their sum is not accurately represented by the sum of their maximas. Finding a more accurate maximum for the combined quantity requires the identification of components of RESPONSE_TIME.

[0027] As specified in p1394a, RESPONSE_TIME includes the time a responding node takes to repeat the received packet and then drive a subsequent arbitration indication. (Note that by examination of the C code, RESPONSE_TIME is defined to include the time it takes to repeat a packet even if the PHY in question is a leaf node.) FIG. 3 illustrates the sequence PHY Y will follow in responding to a received packet. i_Y denotes the timings as seen/interpreted by the PHY state machine. Note that P_Y can be any repeating port on PHY Y. Consequently, the timing constraints referenced to P_Y in the following analysis must hold worst case for any and all repeating ports.

[0028] Beginning with the first arrival of data at P'_Y (t_3), the elaborated timing sequence for RESPONSE_TIME is:

- t_3 First bit of packet received at point P'_Y
- t_3' First bit of packet repeated at point P_Y. t_3' lags t_3 by PHY_DELAY
- t_4 Last bit of packet received at point P'_Y. t_4 follows t_3 by the length of the packet timed in PHY N's clock domain. DATA_END begins
- t_4' Last bit of packet repeated at point P_Y. t_4' lags t_3' by the length of the packet timed in PHY Y's clock domain. The PHY begins "repeating" DATA_END
- t_5 DATA_END concludes at point P'_Y. t_5 follows t_4 by $DATA\_END\_TIME_{Y-1}^{P_{Y-1}}$
- t_{5a} stop_tx_packet() concludes at point i_Y and the state machines command the PHY ports to stop repeating DATA_END. t_{5a} leads t_5' by any transceiver delay.
- t_5' DATA_END concludes at point P_Y. t_5' follows t_4' by $DATA\_END\_TIME_Y^{P_Y}$
- t_{5b} start_tx_packet() commences at point i_Y and the state machines command the PHY ports to begin driving the first arbitration indication of any response. t_{5b} lags t_{5a} by an IDLE_GAP and an unspecified state machine delay herein called SM_DELAY.
- t_6 PHY Y drives arbitration at points P'_Y. t_6 follows t_{5b} by any transceiver delay.

$$t_{3'} = t_3 + PHY\_DELAY_Y^{P'_Y \to P_Y} \tag{19}$$

$$\begin{aligned} t_{4'} &= t_{3'} + \frac{packet\_length}{packet\_speed \cdot BASERATE_Y} \\ &= t_3 + PHY\_DELAY_Y^{P'_Y \to P_Y} + \frac{packet\_length}{packet\_speed \cdot BASERATE_Y} \end{aligned} \tag{20}$$

$$
\begin{aligned}
t_{5'} &= t_{4'} + DATA\_END\_TIME_Y^{P_Y} \\
&= t_3 + PHY\_DELAY_Y^{P_Y' \to P_Y} + \frac{packet\_length}{packet\_speed \cdot BASERATE_Y} + DATA\_END\_TIME_Y^{P_Y}
\end{aligned} \tag{21}
$$

$$
\begin{aligned}
t_{5a} &= t_{5'} - transceiver\_delay_Y^{P_Y} \\
&= t_3 + PHY\_DELAY_Y^{P_Y' \to P_Y} + \frac{packet\_length}{packet\_speed \cdot BASERATE_Y} + DATA\_END\_TIME_Y^{P_Y} - \\
&\quad transceiver\_delay_Y^{P_Y}
\end{aligned} \tag{22}
$$

$$
\begin{aligned}
t_{5b} &= t_{5a} + IDLE\_GAP_Y + SM\_DELAY_Y \\
&= t_3 + PHY\_DELAY_Y^{P_Y' \to P_Y} + \frac{packet\_length}{packet\_speed \cdot BASERATE_Y} + DATA\_END\_TIME_Y^{P_Y} + \\
&\quad IDLE\_GAP_Y + SM\_DELAY_Y - transceiver\_delay_Y^{P_Y}
\end{aligned} \tag{23}
$$

$$
\begin{aligned}
t_6 &= t_{5b} + transceiver\_delay_Y^{P_Y'} \\
&= t_3 + PHY\_DELAY_Y^{P_Y' \to P_Y} + \frac{packet\_length}{packet\_speed \cdot BASERATE_Y} + DATA\_END\_TIME_Y^{P_Y} + \\
&\quad IDLE\_GAP_Y + SM\_DELAY_Y + transceiver\_delay_Y^{P_Y'} - transceiver\_delay_Y^{P_Y}
\end{aligned} \tag{24}
$$

By definition,

$$
RESPONSE\_TIME_Y^{P_Y'} = t_6 - t_5 \tag{25}
$$

and through substitution:

$$
\begin{aligned}
RESPONSE\_TIME_Y^{P_Y'} = {} & PHY\_DELAY_Y^{P_Y' \to P_Y} + DE\_delta^{[P_Y, P_{Y-1}]} + PPM\_delta^{[Y, Y-1]} + \\
& IDLE\_GAP_Y + SM\_DELAY_Y + \\
& transceiver\_delay_Y^{P_Y'} - transceiver\_delay_Y^{P_Y}
\end{aligned} \tag{26}
$$

As such, the combination of RESPONSE_TIME, DE_delta, and PPM_delta from equation (18) can be represented as:

$$
\begin{aligned}
\begin{bmatrix} RESPONSE\_TIME_Y^{P_Y'} + \\ DE\_delta^{[P_{Y-1}, P_X]} + \\ PPM\_delta^{[Y-1, X]} \end{bmatrix}
&= \begin{bmatrix} PHY\_DELAY_Y^{P_Y' \to P_Y} + DE\_delta^{[P_Y, P_{Y-1}]} + PPM\_delta^{[Y, Y-1]} + \\ IDLE\_GAP_Y + SM\_DELAY_Y + transceiver\_delay_Y^{P_Y'} - \\ transceiver\_delay_Y^{P_Y} + DE\_delta^{[P_{Y-1}, P_X]} + PPM\_delta^{[Y-1, X]} \end{bmatrix} \\
&= \begin{bmatrix} PHY\_DELAY_Y^{P_Y' \to P_Y} + DE\_delta^{[P_Y, P_X]} + PPM\_delta^{[Y, X]} + \\ IDLE\_GAP_Y + SM\_DELAY_Y + transceiver\_delay_Y^{P_Y'} - \\ transceiver\_delay_Y^{P_Y} \end{bmatrix}
\end{aligned} \tag{27}
$$

Noting that if PHYs X and Y-1 both adhere to the same minimum timing requirement for DATA_END_TIME and maximum timing requirement for BASE_RATE, then

$$
\begin{aligned}
DE\_delta_{\max}^{[P_Y, P_X]} &= DE\_delta_{\max}^{[P_Y, P_{Y-1}]} \\
PPM\_delta_{\max}^{[Y, X]} &= PPM\_delta_{\max}^{[Y, Y-1]}
\end{aligned} \tag{28}
$$

The combined maximum can be rewritten as:

$$\begin{bmatrix} RESPONSE\_TIME_{Y}^{P_Y'} + \\ DE\_delta^{[P_{Y-1}, P_X]} + \\ PPM\_delta^{[Y-1, X]} \end{bmatrix}_{\max} = \begin{bmatrix} PHY\_DELAY_{Y,\max}^{P_Y' \to P_Y} + DE\_delta_{\max}^{[P_Y, P_{Y-1}]} + PPM\_delta_{\max}^{[Y, Y-1]} + \\ IDLE\_GAP_{Y,\max} + SM\_DELAY_{Y,\max} + \\ transceiver\_delay_{Y,\max}^{P_Y'} - transceiver\_delay_{Y,\min}^{P_Y} \end{bmatrix} \quad (29)$$

Comparing to equation (26) allows

$$\begin{bmatrix} RESPONSE\_TIME_{Y}^{P_Y'} + \\ DE\_delta^{[P_{Y-1}, P_X]} + \\ PPM\_delta^{[Y-1, X]} \end{bmatrix}_{\max} = RESPONSE\_TIME_{Y,\max}^{P_Y'} \quad (30)$$

Finally:

$$gap\_count > \frac{BASERATE_{X,\max} \cdot \begin{bmatrix} Round\_Trip\_Delay_{\max}^{[P_X \, O \, P_Y]} + \\ RESPONSE\_TIME_{Y,\max}^{P_Y'} + \\ PHY\_DELAY_{X,\max}^{P_X} \end{bmatrix} - 27}{16} \quad (31)$$

[0029] For any given topology, the gap count must be set such that subaction gaps observed/generated at one PHY aren't falsely interpreted as arb_reset gaps by another PHY in the network. Subaction gaps are known to be at their largest nearest the PHY that originated the last packet. To ensure that the most recent originating PHY doesn't begin a new fairness interval before all PHYs exit the current one, its arb_reset_gap timeout must be greater than the largest subaction_gap which can legally occur. FIG. 4 illustrates the case in which PHY X originated the most recent packet and PHY Y is responding after a subaction gap with arbitration for the current fairness interval.

[0030] For all topologies, the idle time observed at point P_x must not exceed the arbitration reset gap detection time:

$$Idle_{\max}^{P_X} < arb\_reset\_gap_{\min}^{P_X} \tag{32}$$

[0031] The analysis is identical to the case in which Ack and Iso gaps are preserved with the exception that PHY Y takes longer to respond to the trailing edge of DATA_END. Let PHY Y have a response time of subaction_response_time. Then,

$$\begin{aligned} Idle^{P_X} = {} & Round\_Trip\_Delay^{[P_X \mathrm{O} P_Y]} + subaction\_response\_time_{Y}^{P_Y'} + \\ & DE\_delta^{[P_{Y-1}, P_X]} + PPM\_delta^{[Y-1, X]} \end{aligned} \tag{33}$$

[0032] Substituting into Equation (32), subaction gaps are preserved network-wide if and only if:

$$\begin{bmatrix} Round\_Trip\_Delay^{[P_X \mathrm{O} P_Y]} + subaction\_response\_time_{Y}^{P_Y'} + \\ DE\_delta^{[P_{Y-1}, P_X]} + PPM\_delta^{[Y-1, X]} \end{bmatrix}_{\max} < arb\_reset\_gap_{\min}^{P_X} \tag{34}$$

[0033] The minimum arb_reset_gap at point P_x isn't well known. IEEE1394-1995, in Table 4-33, defines the minimum arb_reset_gap timeout used at a PHY's internal state machines, not at the external interface. It has been argued that the internal and external representations of time may differ by as much as ARB_RESPONSE_DELAY when a PHY is counting elapsed time between an internally generated event and an externally received event. However, the ARB_RESPONSE_DELAY value for a particular PHY isn't generally known externally. Fortunately, the ARB_RESPONSE_DELAY value for a PHY whose FIFO is known to be empty is bounded by the worst case PHY_DELAY reported within the PHY register map. This suggests a realistic bound for the minimum subaction_gap referenced at point P_x:

$$arb\_reset\_gap_{\min}^{P_X} \geq arb\_reset\_gap_{\min}^{i_X} - PHY\_DELAY_{X,\max}^{P_X} \qquad (35)$$

where

$$arb\_reset\_gap_{\min}^{i_X} = \frac{51 + gap\_count \cdot 32}{BASERATE_{X,\max}} \qquad (36)$$

[0034] The maximum subaction_response_time for PHY Y parallels the earlier dissection of RESPONSE_TIME. The timing sequence for subaction_response_time is identical to that of RESPONSE_TIME except that PHY Y, after concluding stop_tx_Packet(), must wait to detect a subaction gap and then wait an additional arb_delay before calling start_tx_packet(). Said differently, the idle period timed internally is a subaction gap plus arb_delay rather than an IDLE_GAP. Consequently, t_{5b} becomes:

$$t_{5b} = t_{5a} + subaction\_gap^{i_Y} + arb\_delay^{i_Y} + SM\_DELAY_Y \tag{37}$$

and

$$subaction\_response\_time_Y^{P_Y'} = RESPONSE\_TIME_Y^{P_Y'} - IDLE\_GAP_Y + subaction\_gap^{i_Y} + arb\_delay^{i_Y} \tag{38}$$

Substituting into Equation (34),

$$\begin{bmatrix} Round\_Trip\_Delay^{[P_X \, 0 \, P_Y]} + RESPONSE\_TIME_Y^{P_Y'} + \\ DE\_delta^{[P_{Y-1}, P_X]} + PPM\_delta^{[Y-1, X]} + \\ subaction\_gap^{i_Y} + arb\_delay^{i_Y} - IDLE\_GAP_Y \end{bmatrix}_{\max} < arb\_reset\_gap_{\min}^{P_X} \tag{39}$$

[0035] Again, RESPONSE_TIME, DE_delta, and PPM_delta are not independent parameters. As shown previously, if PHYs X and Y-1 adhere to the same timing constant limits, the explicit DE_Delta and PPM_delta terms can be subsumed within RESPONSE_TIME giving:

$$\begin{bmatrix} Round\_Trip\_Delay_{\max}^{[P_X \, 0 \, P_Y]} + RESPONSE\_TIME_{Y,\max}^{P_Y'} + \\ subaction\_gap_{\max}^{i_Y} + arb\_delay_{\max}^{i_Y} - MIN\_IDLE\_TIME_Y \end{bmatrix} < arb\_reset\_gap_{\min}^{P_X} \tag{40}$$

where

$$subaction\_gap_{\max}^{i_Y} = \frac{29 + gap\_count \cdot 16}{BASERATE_{Y,\min}} \tag{41}$$

$$arb\_delay_{\max}^{i_Y} = \frac{gap\_count \cdot 4}{BASERATE_{Y,\min}} \tag{42}$$

and

$$IDLE\_GAP_{Y,\min} = MIN\_IDLE\_TIME_Y \tag{43}$$

Combining Equations (35), (36), (40), (41), and (42):

$$\begin{bmatrix} Round\_Trip\_Delay_{\max}^{[P_X \circ P_Y]} + \\ RESPONSE\_TIME_{Y,\max}^{P_Y'} - \\ MIN\_IDLE\_TIME_Y + \\ \dfrac{29 + gap\_count \cdot 20}{BASERATE_{Y,\min}} \end{bmatrix} < \left[\frac{51 + gap\_count \cdot 32}{BASERATE_{X,\max}} - PHY\_DELAY_{X,\max}^{P_X} \right] \tag{44}$$

Solving for gap_count:

$$gap\_count > \frac{BASERATE_{X,\max} \cdot \begin{bmatrix} Round\_Trip\_Delay_{\max}^{[P_X \mathrm{O} P_Y]} + \\ RESPONSE\_TIME_{Y,\max}^{P'_Y} - \\ MIN\_IDLE\_TIME_Y + \\ PHY\_DELAY_{X,\max}^{P_X} \end{bmatrix} + 29 \cdot \frac{BASERATE_{X,\max}}{BASERATE_{Y,\min}} - 51}{32 - 20 \cdot \frac{BASERATE_{X,\max}}{BASERATE_{Y,\min}}} \tag{45}$$

[0036] For any given topology, the gap count must be set such that if a subaction gap is observed following an isochronous packet at one PHY, it is observed at all PHYs. The danger occurs when a subsequent arbitration indication is transmitted in the same direction as the previous data packet. Given that arbitration indications may propagate through intervening PHYs faster than data bits, gaps may be shortened as they are repeated. FIG. 5 illustrates the case in which PHY X originates an isochronous packet, observes a subaction_gap, and begins to drive an arbitration indication.

[0037] For all topologies, the minimum idle time observed at point P'_Y must always exceed the maximum subaction gap detection time:

$$Idle_{\min}^{P'_Y} > subaction\_gap_{\max}^{P'_Y} \tag{46}$$

[0038] The time events t_0 through t_5 are identical to the previous analyses. In this scenario, t_6 follows t_2 by the time it takes PHY X to time subaction_gap and arb_delay:

$$t_6 = t_2 + subaction\_gap^{P_X} + arb\_delay^{P_X}$$
$$= t_0 + \frac{packet\_length}{packet\_speed \cdot BASERATE_X} + DATA\_END\_TIME_X^{P_X} + subaction\_gap^{P_X} + arb\_delay^{P_X} \quad (47)$$

[0039] The 1995 specification provides the timeouts used internally by the state machine. The externally observed timing requirements could differ (given possible mismatches in transceiver delay and state machines between the leading edge of IDLE and the leading edge of the subsequent arbitration indication). However, previous works have suggested any such delays could and should be well matched and that the external timing would follow the internal timing exactly. Consequently,

$$subaction\_gap^{P_X} + arb\_delay^{P_X} = subaction\_gap^{i_X} + arb\_delay^{i_X} \quad (48)$$

T7 follows T6 by the time it takes the arbitration signal to propagate through the intervening PHYs and cables:

$$t_7 = t_6 + \sum_{n=X+1}^{Y-1}\left(cable\_delay_n + ARB\_RESPONSE\_DELAY_n^{P_n' \to P_n}\right) + cable\_delay_X$$
$$= t_0 + \frac{packet\_length}{packet\_speed \cdot BASERATE_X} + DATA\_END\_TIME_X^{P_X} + subaction\_gap^{i_X} + arb\_delay^{i_X} + \sum_{n=X+1}^{Y-1}\left(cable\_delay_n + ARB\_RESPONSE\_DELAY_n^{P_n' \to P_n}\right) + cable\_delay_X \quad (49)$$

Given t_0 through t_7 above, the Idle time seen at point P'_Y is given as:

$$\begin{aligned} Idle^{P'_Y} &= t_7 - t_5 \\ &= subaction\_gap^{i_X} + arb\_delay^{i_X} - \\ &\sum_{n=X+1}^{Y-1}\left(PHY\_DELAY_n^{P'_n \to P_n} - ARB\_RESPONSE\_DELAY_n^{P'_n \to P_n}\right) - \\ &DE\_delta^{[P_{Y-1}, P_X]} - PPM\_delta^{[Y-1,X]} \end{aligned} \tag{50}$$

Let

$$Data\_Arb\_Mismatch^{[P_X \to P_Y]} = \sum_{n=X+1}^{Y-1}\left(PHY\_DELAY_n^{P'_n \to P_n} - ARB\_RESPONSE\_DELAY_n^{P'_n \to P_n}\right) \tag{51}$$

Then,

$$\begin{aligned} Idle^{P'_Y} &= t_7 - t_5 \\ &= subaction\_gap^{i_X} + arb\_delay^{i_X} - Data\_Arb\_Mismatch^{[P_X \to P_Y]} - \\ &DE\_delta^{[P_{Y-1}, P_X]} - PPM\_delta^{[Y-1,X]} \end{aligned} \tag{52}$$

[0040] For the maximum subaction_gap detection time at point P'_Y, the 1995 standard again only specifies the internal state machine timeout values. FIG. 6 provides the timing reference for relating the external gap detection times to the

internal ones. The elaborated timing sequence is identical to the case for RESPONSE_TIME through point t_5'. The remaining sequence is:

T_7 The arbitration indication launched by PHY X arrives at point P'_Y

T_{7a} The arbitration indication launched by PHY X arrives at point iY. t_{7a} lags t_7 by an unspecified arbitration detection time, herein termed ARB_DETECTION_TIME

The externally seen gap at point P'_Y is given as

$$gap^{P'_Y} = t_7 - t_5 \tag{53}$$

The corresponding internal gap at point iY is

$$gap^{i_Y} = t_{7a} - t_{5a} \tag{54}$$

Given that

$$t_{7a} = t_7 + ARB\_DETECTION\_TIME_Y^{P'_Y} \tag{55}$$

the external gap can be expressed as

$$\begin{aligned} gap^{P'_Y} &= t_7 - t_5 \\ &= t_{7a} - t_5 - ARB\_DETECTION\_TIME_Y^{P'_Y} \\ &= t_{7a} - t_{5a} + t_{5a} - t_5 - ARB\_DETECTION\_TIME_Y^{P'_Y} \\ &= gap^{i_Y} + t_{5a} - t_5 - ARB\_DETECTION\_TIME_Y^{P'_Y} \\ &= gap^{i_Y} + PHY\_DELAY_Y^{P'_Y \to P_Y} + DE\_delta^{[P_Y, P_{Y-1}]} + PPM\_delta^{[Y,Y-1]} - \\ &\quad transceiver\_delay_Y^{P_Y} - ARB\_DETECTION\_TIME_Y^{P'_Y} \end{aligned} \tag{56}$$

Consequently,

$$\begin{aligned} subaction\_gap^{P'_Y} = subaction\_gap^{i_Y} &+ PHY\_DELAY_Y^{P'_Y \to P_Y} + \\ &DE\_delta^{[P_Y, P_{Y-1}]} + PPM\_delta^{[Y,Y-1]} - \\ &transceiver\_delay_Y^{P_Y} - ARB\_DETECTION\_TIME_Y^{P'_Y} \end{aligned} \tag{57}$$

Substituting (52) and (57) into (46) yields

$$\begin{bmatrix} subaction\_gap^{i_X} + arb\_delay^{i_X} - \\ Data\_Arb\_Mismatch^{[P_X \to P_Y]} - \\ DE\_delta^{[P_{Y-1}, P_X]} - PPM\_delta^{[Y-1,X]} \end{bmatrix} > \begin{bmatrix} subaction\_gap^{i_Y} + PHY\_DELAY_Y^{P'_Y \to P_Y} + \\ DE\_delta^{[P_Y, P_{Y-1}]} + PPM\_delta^{[Y,Y-1]} - \\ transceiver\_delay_Y^{P_Y} - ARB\_DETECTION\_TIME_Y^{P'_Y} \end{bmatrix} \tag{58}$$

The inequality holds generally if

$$\left[subaction\_gap^{i_X} + arb\_delay^{i_X}\right]_{min} > \begin{bmatrix} subaction\_gap^{i_Y} + PHY\_DELAY_Y^{P_Y' \to P_Y} + \\ DE\_delta^{[P_Y, P_{Y-1}]} + PPM\_delta^{[Y, Y-1]} + \\ DE\_delta^{[P_{Y-1}, P_X]} + PPM\_delta^{[Y-1, X]} + \\ Data\_Arb\_Mismatch^{[P_X \to P_Y]} - \\ transceiver\_delay_Y^{P_Y} - ARB\_DETECTION\_TIME_Y^{P_Y'} \end{bmatrix}_{max} \quad (59)$$

Combining the DE_Delta and PPM_delta terms gives:

$$\left[subaction\_gap^{i_X} + arb\_delay^{i_X}\right]_{min} > \begin{bmatrix} subaction\_gap^{i_Y} + PHY\_DELAY_Y^{P_Y' \to P_Y} + \\ DE\_delta^{[P_Y, P_X]} + PPM\_delta^{[Y, X]} + \\ Data\_Arb\_Mismatch^{[P_X \to P_Y]} - \\ transceiver\_delay_Y^{P_Y} - ARB\_DETECTION\_TIME_Y^{P_Y'} \end{bmatrix}_{max} \quad (60)$$

By assuming

$$DE\_delta^{[P_Y, P_X]} + PPM\_delta^{[Y, X]} \leq transceiver\_delay_Y^{P_Y} + ARB\_DETECTION\_TIME_Y^{P_Y'} \quad (61)$$

the constraining inequality can be further simplified to give

$$\left[subaction\_gap_{min}^{i_X} + arb\_delay_{min}^{i_X}\right] > \begin{bmatrix} subaction\_gap_{max}^{i_Y} + PHY\_DELAY_{Y,max}^{P_Y' \to P_Y} + \\ Data\_Arb\_Mismatch_{max}^{[P_X \to P_Y]} \end{bmatrix} \quad (62)$$

where

$$subaction\_gap_{\min}^{i_X} = \frac{27 + gap\_count \cdot 16}{BASERATE_{X,\max}} \tag{63}$$

$$arb\_delay_{\min}^{i_X} = \frac{gap\_count \cdot 4}{BASERATE_{X,\max}} \tag{64}$$

and

$$subaction\_gap_{\max}^{i_Y} = \frac{29 + gap\_count \cdot 16}{BASERATE_{Y,\min}} \tag{65}$$

Solving for gap count,

$$gap\_count > \frac{BASERATE_{X,\max} \cdot \begin{bmatrix} PHY\_DELAY_{Y,\max}^{P_Y' \to P_Y} + \\ Data\_Arb\_Mismatch_{\max}^{[P_X \to P_Y]} \end{bmatrix} + 29 \cdot \dfrac{BASERATE_{X,\max}}{BASERATE_{Y,\min}} - 27}{20 - 16 \cdot \dfrac{BASERATE_{X,\max}}{BASERATE_{Y,\min}}} \tag{66}$$

[0041] For any given topology, the gap count must be set such that if an arbitration reset gap is observed following an asynchronous packet at one PHY, it is observed at all PHYs. The danger occurs when a subsequent arbitration indication is transmitted in the same direction as the previous data packet. Given that arbitration indications may propagate through intervening PHYs faster than data bits, gaps may be shortened as they are repeated. FIG. 7 illustrates the case in which PHY X originates an asynchronous packet, observes an arbitration reset gap, and begins to drive an arbitration indication.

[0042] For all topologies, the minimum idle time observed at point P'_Y must always exceed the maximum arbitration reset gap detection time:

$$Idle_{\min}^{P'_Y} > arb\_reset\_gap_{\max}^{P'_Y} \tag{67}$$

[0043] The time events t_0 through t_5 are identical to the previous analyses. In this scenario, t_6 follows t_2 by the time it takes PHY X to time arb_reset_gap and arb_delay:

$$\begin{aligned} t_6 &= t_2 + arb\_reset\_gap^{P_X} + arb\_delay^{P_X} \\ &= t_0 + \frac{packet\_length}{packet\_speed \cdot BASERATE_X} + DATA\_END\_TIME_X^{P_X} + \\ &\quad arb\_reset\_gap^{P_X} + arb\_delay^{P_X} \end{aligned} \tag{68}$$

[0044] The 1995 IEEE 1394 standard provides the timeouts used internally by the state machine. The externally observed timing requirements could differ (given possible mismatches in transceiver delay and state machines between the leading edge of IDLE and the leading edge of the subsequent arbitration indication). However, previous works have suggested any such delays could and should be well matched and that the external timing would follow the internal timing exactly. Consequently,

$$arb\_reset\_gap^{P_X} + arb\_delay^{P_X} = arb\_reset\_gap^{i_X} + arb\_delay^{i_X} \tag{69}$$

[0045] T7 follows T6 by the time it takes the arbitration signal to propagate through the intervening PHYs and cables:

$$\begin{aligned} t_7 &= t_6 + \sum_{n=X+1}^{Y-1}\left(cable\_delay_n + ARB\_RESPONSE\_DELAY_n^{P_n' \to P_n}\right) + cable\_delay_X \\ &= t_0 + \frac{packet\_length}{packet\_speed \cdot BASERATE_X} + DATA\_END\_TIME_X^{P_X} + \\ &\quad arb\_reset\_gap^{i_X} + arb\_delay^{i_X} + \\ &\quad \sum_{n=X+1}^{Y-1}\left(cable\_delay_n + ARB\_RESPONSE\_DELAY_n^{P_n' \to P_n}\right) + cable\_delay_X \end{aligned} \tag{70}$$

[0046] Given t_0 through t_7 above, the Idle time seen at point P'_Y is given as:

$$Idle^{P'_Y} = t_7 - t_5$$
$$= arb\_reset\_gap^{i_X} + arb\_delay^{i_X} - Data\_Arb\_Mismatch^{[P_X \to P_Y]} - DE\_delta^{[P_{Y-1}, P_X]} - PPM\_delta^{[Y-1, X]} \quad (71)$$

[0047] For the maximum arbitration_reset_gap detection time at point P'_Y, equation (56) gives:

$$arb\_reset\_gap^{P'_Y} = arb\_reset\_gap^{i_Y} + PHY\_DELAY_Y^{P'_Y \to P_Y} + DE\_delta^{[P_Y, P_{Y-1}]} + PPM\_delta^{[Y, Y-1]} - transceiver\_delay_Y^{P_Y} - ARB\_DETECTION\_TIME_Y^{P'_Y} \quad (72)$$

[0048] Substituting (71) and (72) into (67) yields

$$\begin{bmatrix} arb\_reset\_gap^{i_X} + arb\_delay^{i_X} - \\ Data\_Arb\_Mismatch^{[P_X \to P_Y]} - \\ DE\_delta^{[P_{Y-1}, P_X]} - PPM\_delta^{[Y-1, X]} \end{bmatrix} > \begin{bmatrix} arb\_reset\_gap^{i_Y} + PHY\_DELAY_Y^{P'_Y \to P_Y} + \\ DE\_delta^{[P_Y, P_{Y-1}]} + PPM\_delta^{[Y, Y-1]} - \\ transceiver\_delay_Y^{P_Y} - ARB\_DETECTION\_TIME_Y^{P'_Y} \end{bmatrix} \quad (73)$$

[0049] The inequality holds generally if

$$\left[arb\_reset\_gap^{i_X} + arb\_delay^{i_X}\right]_{min} > \begin{bmatrix} arb\_reset\_gap^{i_Y} + PHY\_DELAY_Y^{P_Y' \to P_Y} + \\ DE\_delta^{[P_Y, P_{Y-1}]} + PPM\_delta^{[Y, Y-1]} + \\ DE\_delta^{[P_{Y-1}, P_X]} + PPM\_delta^{[Y-1, X]} + \\ Data\_Arb\_Mismatch^{[P_X \to P_Y]} - \\ transceiver\_delay_Y^{P_Y} - ARB\_DETECTION\_TIME_Y^{P_Y'} \end{bmatrix}_{max} \tag{74}$$

[0050] Combining the DE_Delta and PPM_delta terms gives:

$$\left[arb\_reset\_gap^{i_X} + arb\_delay^{i_X}\right]_{min} > \begin{bmatrix} arb\_reset\_gap^{i_Y} + PHY\_DELAY_Y^{P_Y' \to P_Y} + \\ DE\_delta^{[P_Y, P_X]} + PPM\_delta^{[Y, X]} + \\ Data\_Arb\_Mismatch^{[P_X \to P_Y]} - \\ transceiver\_delay_Y^{P_Y} - ARB\_DETECTION\_TIME_Y^{P_Y'} \end{bmatrix}_{max} \tag{75}$$

[0051] By requiring

$$DE\_delta^{[P_Y, P_X]} + PPM\_delta^{[Y, X]} \leq transceiver\_delay_Y^{P_Y} + ARB\_DETECTION\_TIME_Y^{P_Y'}{}_Y \tag{76}$$

[0052] the constraining inequality can be further simplified to give

$$\left[arb\_reset\_gap_{min}^{i_X} + arb\_delay_{min}^{i_X}\right] > \begin{bmatrix} arb\_reset\_gap_{max}^{i_Y} + PHY\_DELAY_{Y,max}^{P_Y' \to P_Y} + \\ Data\_Arb\_Mismatch_{max}^{[P_X \to P_Y]} \end{bmatrix} \tag{77}$$

where

$$arb\_reset\_gap_{\min}^{i_X} = \frac{51 + gap\_count \cdot 32}{BASERATE_{X,\max}} \tag{78}$$

$$arb\_delay_{\min}^{i_X} = \frac{gap\_count \cdot 4}{BASERATE_{X,\max}} \tag{79}$$

and

$$arb\_reset\_gap_{\max}^{i_Y} = \frac{53 + gap\_count \cdot 32}{BASERATE_{Y,\min}} \tag{80}$$

[0053] Solving for gap count,

$$gap\_count > \frac{BASERATE_{X,\max} \cdot \left[\begin{array}{l} PHY\_DELAY_{Y,\max}^{P_Y' \to P_Y} + \\ Data\_Arb\_Mismatch_{\max}^{[P_X \to P_Y]} \end{array} \right] + 53 \cdot \frac{BASERATE_{X,\max}}{BASERATE_{Y,\min}} - 51}{36 - 32 \cdot \frac{BASERATE_{X,\max}}{BASERATE_{Y,\min}}} \tag{81}$$

[0054] Equations (31), (45), (66) and (81) place a lower bound on gap count. Let:

$$gap\_count_A = \frac{BASERATE_{X,\max} \cdot \begin{bmatrix} Round\_Trip\_Delay_{\max}^{[P_X \,\mathrm{O}\, P_Y]} + \\ RESPONSE\_TIME_{Y,\max}^{P'_Y} + \\ PHY\_DELAY_{X,\max}^{P_X} \end{bmatrix} - 27}{16} \tag{82}$$

$$gap\_count_B = \frac{BASERATE_{X,\max} \cdot \begin{bmatrix} Round\_Trip\_Delay_{\max}^{[P_X \,\mathrm{O}\, P_Y]} + \\ RESPONSE\_TIME_{Y,\max}^{P'_Y} - \\ MIN\_IDLE\_TIME_Y + \\ PHY\_DELAY_{X,\max}^{P_X} \end{bmatrix} + 29 \cdot \frac{BASERATE_{X,\max}}{BASERATE_{Y,\min}} - 51}{32 - 20 \cdot \frac{BASERATE_{X,\max}}{BASERATE_{Y,\min}}} \tag{83}$$

$$gap\_count_C = \frac{BASERATE_{X,\max} \cdot \begin{bmatrix} Data\_Arb\_Mismatch_{\max}^{[P_X \to P_Y]} + \\ PHY\_DELAY_{Y,\max}^{P'_Y \to P_Y} \end{bmatrix} + 29 \cdot \frac{BASERATE_{X,\max}}{BASERATE_{Y,\min}} - 27}{20 - 16 \cdot \frac{BASERATE_{X,\max}}{BASERATE_{Y,\min}}} \tag{84}$$

$$gap\_count_D = \frac{BASERATE_{X,\max} \cdot \begin{bmatrix} Data\_Arb\_Mismatch_{\max}^{[P_X \to P_Y]} + \\ PHY\_DELAY_{Y,\max}^{P'_Y \to P_Y} \end{bmatrix} + 53 \cdot \frac{BASERATE_{X,\max}}{BASERATE_{Y,\min}} - 51}{36 - 32 \cdot \frac{BASERATE_{X,\max}}{BASERATE_{Y,\min}}} \tag{85}$$

[0055] Given the ratio of maximum to minimum BASERATE is always >1 and that MIN_IDLE_TIME is ~40 ns, it is clear that:

$$gap\_count_B > gap\_count_A \tag{86}$$

and

$$gap\_count_D > gap\_count_C \tag{87}$$

[0056] To select an appropriate gap count for a given topology, both gap_count_B and gap_count_D must be calculated, rounded up to the next integer, and the maximum of the two results selected.

[0057] For IEEE1394-1995 style topologies (assumed to be limited to 4.5m cables and a worst case PHY_DELAY of 144 ns), a table can be constructed to provide the gap count setting as a function of hops. In constructing such a table, the constant values in Table 2 are assumed.

Table 2: PHY Timing Constants

Parameter	**Minimum**	**Maximum**
ARB_RESPONSE_DELAY[1]	PHY_DELAY(max) – 60 ns	PHY_DELAY(max)
BASERATE	98.294 mbps	98.314 mbps
cable_delay		22.725 ns
MIN_IDLE_TIME	40 ns	
PHY_DELAY		144 ns

RESPONSE_TIME		PHY_DELAY + 100 ns

[0058] The resulting gap count versus Cable Hops can then be calculated:

Table 3 : Gap Count as a function of hops

Hops	Gap Count
1	5
2	7
3	8
4	10
5	13
6	16
7	18
8	21
9	24
10	26
11	29
12	32
13	35
14	37
15	40
16	43
17	46
18	48
19	51
20	54
21	57
22	59
23	62

[0059] Pinging provides an effective way to set an optimal gap count for topologies with initially unspecified or unknown PHY or cable delays. Specifically, pinging allows determination of an instantaneous Round_Trip_Delay between two given points. Once the worst case Round_Trip Delay has been

determined via pinging, gap_count_b and gap_count_d can be calculated and the appropriate gap count selected.

[0060] The Jitter value specified in the PHY register map was introduced to help relate instantaneous measurements of ROUND_TRIP_DELAY to the maximum possible ROUND_TRIP_DELAY between two points. Specifically, the outbound PHY_DELAY and return ARB_RESPONSE_DELAY measured between a given ordered pair of ports on a PHY (say P_c out to and back from P_d) can be related to the maximum outbound PHY_DELAY and return ARB_RESPONSE_DELAY between <u>any and all</u> ordered pairs of ports (referenced as P_a & P_b) on the same PHY:

$$0 \leq \left[\frac{PHY\_DELAY_{n,\max}^{P_a \to P_b} + ARB\_RESPONSE\_DELAY_{n,\max}^{P_b \to P_a}}{2} - \frac{PHY\_DELAY_{n,meas}^{P_c \to P_d} + ARB\_RESPONSE\_DELAY_{n,meas}^{P_d \to P_c}}{2} \right] \leq Jitter_n \tag{88}$$

[0061] Noting that a measured value can never exceed a maximum value between order ports, the following corollary relating two independent measurements can be proven for any and all combination of ordered ports:

$$\left| \frac{PHY\_DELAY_{n,meas_1}^{P_a \to P_b} + ARB\_RESPONSE\_DELAY_{n,meas_1}^{P_b \to P_a}}{2} - \frac{PHY\_DELAY_{n,meas_2}^{P_c \to P_d} + ARB\_RESPONSE\_DELAY_{n,meas_2}^{P_d \to P_c}}{2} \right| \leq Jitter_n \tag{89}$$

[0062] In order for a bus manager to calculate ordered leaf-to-leaf delays via a series of ping requests launched from the bus manager, a number of ROUND_TRIP_DELAY relationships will be required and are derived below.

$$Round\_Trip\_Delay_{\max}^{[P_X \mathrm{O} P_Y]}$$

[0063] Using the definition of Round_Trip_Delay first provided in equation (12) as guidance, the roundtrip delay between Nodes X and Y from the perspective of Node X can be written as:

$$Round\_Trip\_Delay_{\max}^{[P_X \mathrm{O} P_Y]} = \sum_{n=X+1}^{Y-1} \begin{pmatrix} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,\max}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,\max}^{P_n \to P_n'} \end{pmatrix} + 2 \cdot cable\_delay_X \tag{90}$$

[0064] From equation (88), the maximum PHY_DELAY and ARB_RESPONSE_DELAY between an ordered pair of ports can be bounded by the measured delays plus the overall jitter sum yielding:

$$Round\_Trip\_Delay_{\max}^{[P_X \mathrm{O} P_Y]} \leq \sum_{n=X+1}^{Y-1} \begin{pmatrix} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,meas_l}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas_l}^{P_n \to P_n'} + \\ 2 \cdot Jitter_n \end{pmatrix} + 2 \cdot cable\_delay_X \tag{91}$$

[0065] Comparison to the definition of Round_Trip_Delay then allows

$$Round\_Trip\_Delay_{\max}^{[P_X \text{O} P_Y]} \leq Round\_Trip\_Delay_{meas_I}^{[P_X \text{O} P_Y]} + \sum_{n=X+1}^{Y-1} 2 \cdot Jitter_n \tag{92}$$

$$Round\_Trip\_Delay_{\max}^{[P_Y \text{O} P_X]}$$

[0066] Using the definition of Round_Trip_Delay first provided in equation (12) as guidance, the roundtrip delay between Nodes X and Y from the perspective of Node Y can be written as:

$$Round\_Trip\_Delay_{\max}^{[P_Y \text{O} P_X]} = \sum_{n=X+1}^{Y-1} \begin{pmatrix} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,\max}^{P_n \to P_n'} + \\ ARB\_RESPONSE\_DELAY_{n,\max}^{P_n' \to P_n} \end{pmatrix} + \\ 2 \cdot cable\_delay_X \tag{93}$$

[0067] From equation (88), the maximum PHY_DELAY and ARB_RESPONSE_DELAY between an ordered pair of ports can be related to the measured delays observed in the reverse direction:

$$\begin{pmatrix} PHY\_DELAY_{n,\max}^{P_n \to P_n'} + \\ ARB\_RESPONSE\_DELAY_{n,\max}^{P_n' \to P_n} \end{pmatrix} \leq 2 \cdot Jitter_n + \begin{pmatrix} PHY\_DELAY_{n,meas_I}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas_I}^{P_n \to P_n'} \end{pmatrix} \tag{94}$$

[0068] allowing the maximum round trip between Nodes X and Y to be rewritten as:

$$Round\_Trip\_Delay_{\max}^{[P_Y \circ P_X]} \leq \sum_{n=X+1}^{Y-1} \begin{pmatrix} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,meas_l}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas_l}^{P_n \to P_n'} + \\ 2 \cdot Jitter_n \end{pmatrix} + \tag{95}$$
$$2 \cdot cable\_delay_X$$

[0069] Comparison to the definition of Round_Trip_Delay then allows

$$Round\_Trip\_Delay_{\max}^{[P_Y \circ P_X]} \leq Round\_Trip\_Delay_{meas_l}^{[P_X \circ P_Y]} + \sum_{n=X+1}^{Y-1} 2 \cdot Jitter_n \tag{96}$$

$$Round\_Trip\_Delay_{\max}^{[P_N \circ P_Y]}$$

[0070] Using the definition of Round_Trip_Delay first provided in equation (12) as guidance, the roundtrip delay between Nodes N and Y from the perspective of Node N can be written as:

$$Round\_Trip\_Delay_{\max}^{[P_N \circ P_Y]} = \sum_{n=N+1}^{Y-1} \begin{pmatrix} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,\max}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,\max}^{P_n \to P_n'} \end{pmatrix} + \tag{97}$$
$$2 \cdot cable\_delay_N$$

[0071] From equation (88), the maximum PHY_DELAY and ARB_RESPONSE_DELAY between an ordered pair of ports can be bounded by the measured delays plus the overall jitter sum yielding:

$$Round\_Trip\_Delay_{\max}^{[P_N \circ P_Y]} \leq \sum_{n=N+1}^{Y-1} \begin{pmatrix} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,meas_l}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas_l}^{P_n \to P_n'} + \\ 2 \cdot Jitter_n \end{pmatrix} + \\ 2 \cdot cable\_delay_N \tag{98}$$

[0072] Introducing offsetting terms to the right side:

$$\begin{aligned} Round\_Trip\_Delay_{\max}^{[P_N \circ P_Y]} \leq & \sum_{n=X+1}^{Y-1} \begin{pmatrix} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,meas_l}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas_l}^{P_n \to P_n'} + \\ 2 \cdot Jitter_n \end{pmatrix} + \\ & 2 \cdot cable\_delay_X - \\ & \sum_{n=X+1}^{N-1} \begin{pmatrix} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,meas_l}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas_l}^{P_n \to P_n'} + \\ 2 \cdot Jitter_n \end{pmatrix} - \\ & 2 \cdot cable\_delay_X - PHY\_DELAY_{N,meas_l}^{P_N' \to P_N} - \\ & ARB\_RESPONSE\_DELAY_{N,meas_l}^{P_N \to P_N'} - 2 \cdot Jitter_N \end{aligned} \tag{99}$$

[0073] Equations (89) and the fact that measured delays are at no smaller than minimum delays allow simplification to:

$$Round\_Trip\_Delay_{\max}^{[P_N \circ P_Y]} \leq \sum_{n=X+1}^{Y-1}\begin{pmatrix} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,meas_1}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas_1}^{P_n \to P_n'} + \\ 2 \cdot Jitter_n \end{pmatrix} + \\ 2 \cdot cable\_delay_X - \\ \sum_{n=X+1}^{N-1}\begin{pmatrix} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,meas_2}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas_2}^{P_n \to P_n'} \end{pmatrix} - \\ 2 \cdot cable\_delay_X - PHY\_DELAY_{N,\min}^{P_N' \to P_N} - \\ ARB\_RESPONSE\_DELAY_{N,\min}^{P_N \to P_N'} - 2 \cdot Jitter_N \qquad (100)$$

[0074] Comparison to the definition of Round_Trip_Delay then allows

$$Round\_Trip\_Delay_{\max}^{[P_N \circ P_Y]} \leq Round\_Trip\_Delay_{meas_1}^{[P_X \circ P_Y]} + \sum_{n=X+1}^{Y-1} 2 \cdot Jitter_n - \\ Round\_Trip\_Delay_{meas_2}^{[P_X \circ P_N]} - PHY\_DELAY_{N,\min}^{P_N' \to P_N} - \\ ARB\_RESPONSE\_DELAY_{N,\min}^{P_N \to P_N'} - 2 \cdot Jitter_N \qquad (101)$$

$$Round\_Trip\_Delay_{\max}^{[P_Y \circ P_N]}$$

[0075] Using the definition of Round_Trip_Delay first provided in equation (12) as guidance, the roundtrip delay between Nodes N and Y from the perspective of Node Y can be written as:

$$Round\_Trip\_Delay_{\max}^{[P_Y \, \text{O} \, P_N]} = \sum_{n=N+1}^{Y-1} \left(\begin{array}{l} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,\max}^{P_n \rightarrow P_n'} + \\ ARB\_RESPONSE\_DELAY_{n,\max}^{P_n' \rightarrow P_n} \end{array} \right) + 2 \cdot cable\_delay_N \quad (102)$$

[0076] From equation (88), the maximum PHY_DELAY and ARB_RESPONSE_DELAY between an ordered pair of ports can be related to the measured delays observed in the reverse direction:

$$\left(\begin{array}{l} PHY\_DELAY_{n,\max}^{P_n \rightarrow P_n'} + \\ ARB\_RESPONSE\_DELAY_{n,\max}^{P_n' \rightarrow P_n} \end{array} \right) \le 2 \cdot Jitter_n + \left(\begin{array}{l} PHY\_DELAY_{n,meas_I}^{P_n' \rightarrow P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas_I}^{P_n \rightarrow P_n'} \end{array} \right) \quad (103)$$

[0077] allowing the maximum round trip between Nodes N and Y to be rewritten as:

$$Round\_Trip\_Delay_{\max}^{[P_Y \, \text{O} \, P_N]} \le \sum_{n=N+1}^{Y-1} \left(\begin{array}{l} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,meas_I}^{P_n' \rightarrow P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas_I}^{P_n \rightarrow P_n'} + \\ 2 \cdot Jitter_n \end{array} \right) + 2 \cdot cable\_delay_N \quad (104)$$

[0078] Introducing offsetting terms to the right side:

$$Round\_Trip\_Delay_{\max}^{[P_Y \circ P_N]} \leq \sum_{n=X+1}^{Y-1} \begin{pmatrix} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,meas_1}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas_1}^{P_n \to P_n'} + \\ 2 \cdot Jitter_n \end{pmatrix} + \\ 2 \cdot cable\_delay_X - \\ \sum_{n=X+1}^{N-1} \begin{pmatrix} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,meas_1}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas_1}^{P_n \to P_n'} + \\ 2 \cdot Jitter_n \end{pmatrix} - \\ 2 \cdot cable\_delay_X - PHY\_DELAY_{N,meas_1}^{P_N' \to P_N} - \\ ARB\_RESPONSE\_DELAY_{N,meas_1}^{P_N \to P_N'} - 2 \cdot Jitter_N \qquad (105)$$

[0079] Equations (89) and the fact that measured delays are at no smaller than minimum delays allow simplification to:

$$Round\_Trip\_Delay_{\max}^{[P_Y \circ P_N]} \leq \sum_{n=X+1}^{Y-1} \begin{pmatrix} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,meas_1}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas_1}^{P_n \to P_n'} + \\ 2 \cdot Jitter_n \end{pmatrix} + \\ 2 \cdot cable\_delay_X - \\ \sum_{n=X+1}^{N-1} \begin{pmatrix} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,meas_2}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas_2}^{P_n \to P_n'} \end{pmatrix} - \\ 2 \cdot cable\_delay_X - PHY\_DELAY_{N,\min}^{P_N' \to P_N} - \\ ARB\_RESPONSE\_DELAY_{N,\min}^{P_N \to P_N'} - 2 \cdot Jitter_N \qquad (106)$$

[0080] Comparison to the definition of Round_Trip_Delay then allows

$$Round\_Trip\_Delay_{\max}^{[P_Y \mathrm{O}\, P_N]} \leq Round\_Trip\_Delay_{meas_1}^{[P_X \mathrm{O}\, P_Y]} + \sum_{n=X+1}^{Y-1} 2 \cdot Jitter_n - Round\_Trip\_Delay_{meas_2}^{[P_X \mathrm{O}\, P_N]} - PHY\_DELAY_{N,\min}^{P_N' \to P_N} - ARB\_RESPONSE\_DELAY_{N,\min}^{P_N \to P_N'} - 2 \cdot Jitter_N \qquad (107)$$

[0081] PHY pinging provides a low level mechanism to directly measure round trip delays between two nodes by timing link initiated subactions. However, pinging does introduce some uncertainty in the measured delay. Any gap count algorithm which employs PHY pinging must compensate for such uncertainty. FIG. 8 depicts a ping subaction issued by the link in Node X and directed to Node Y.

[0082] The timing reference points t_1 through t_7 are identical to those used in the previous gap count derivations. Additionally:

t_1' Coincident with the rising SCLK edge in which the PHY first samples IDLE after a link transmission. t_1' leads t_1 by LINK_TO_BUS_DELAY

t_7' Coincident with the rising SCLK edge in which the PHY is driving the first RECEIVE indication to the link. (The PHY presumably drove RECEIVE off of the previous clock transition.) t_7' lags t_7 by BUS_TO_LINK_DELAY

[0083] The ping time measured by the link (in SCLK cycles) is then given by:

$$
\begin{aligned}
Ping\_Time_{meas}^{[P_X \circ P_Y]} &= t_7' - t_1' \\
&= BUS\_TO\_LINK\_DELAY_{X,meas} + t_7 - t_1 + LINK\_TO\_BUS\_DELAY_{X,meas} \\
&= BUS\_TO\_LINK\_DELAY_{X,meas} + LINK\_TO\_BUS\_DELAY_{X,meas} + \\
&\quad t_0 + \sum_{n=X+1}^{Y-1} \left(\begin{array}{l} 2 \cdot cable\_delay_n + PHY\_DELAY_{n,meas}^{P_n' \to P_n} + \\ ARB\_RESPONSE\_DELAY_{n,meas}^{P_n \to P_n'} \end{array} \right) + \\
&\quad 2 \cdot cable\_delay_X + \frac{packet\_length}{packet\_speed \cdot BASERATE_{Y-1}} + \\
&\quad DATA\_END\_TIME_{Y-1,meas}^{P_{Y-1}} + RESPONSE\_TIME_{Y,meas}^{P_Y'} - \\
&\quad t_0 - \frac{packet\_length}{packet\_speed \cdot BASERATE_X} \\
&= BUS\_TO\_LINK\_DELAY_{X,meas} + LINK\_TO\_BUS\_DELAY_{X,meas} + \\
&\quad Round\_Trip\_Delay_{meas}^{[P_X \circ P_Y]} + PPM\_delta^{[Y-1,X]} + \\
&\quad DATA\_END\_TIME_{Y-1,meas}^{P_{Y-1}} + RESPONSE\_TIME_{Y,meas}^{P_Y'}
\end{aligned} \tag{108}
$$

[0084] Solving for the measured Round_Trip_Delay gives:

$$
\begin{aligned}
Round\_Trip\_Delay_{meas}^{[P_X \circ P_Y]} = {} & Ping\_Time_{meas}^{[P_X \circ P_Y]} - BUS\_TO\_LINK\_DELAY_{X,meas} - \\
& LINK\_TO\_BUS\_DELAY_{X,meas} - PPM\_delta^{[Y-1,X]} - \\
& DATA\_END\_TIME_{Y-1,meas}^{P_{Y-1}} - RESPONSE\_TIME_{Y,meas}^{P_Y'}
\end{aligned} \tag{109}
$$

[0085] Remembering that RESPONSE_TIME (min or max) absorbs PPM_delta, an upper and lower bound can be defined for Round_Trip_Delay:

$$
\begin{aligned}
Round\_Trip\_Delay_{Ping,\max}^{[P_X \circ P_Y]} = {} & Ping\_Time_{meas}^{[P_X \circ P_Y]} - BUS\_TO\_LINK\_DELAY_{X,\min} - \\
& LINK\_TO\_BUS\_DELAY_{X,\min} - DATA\_END\_TIME_{Y-1,\min}^{P_{Y-1}} - \\
& RESPONSE\_TIME_{Y,\min}^{P_Y'}
\end{aligned} \tag{110}
$$

and

$$Round\_Trip\_Delay_{Ping,\min}^{[P_X \circ P_Y]} = Ping\_Time_{meas}^{[P_X \circ P_Y]} - BUS\_TO\_LINK\_DELAY_{X,\max} - LINK\_TO\_BUS\_DELAY_{X,\max} - DATA\_END\_TIME_{Y-1,\max}^{P_{Y-1}} - RESPONSE\_TIME_{Y,\max}^{P_Y'} \tag{111}$$

such that

$$Round\_Trip\_Delay_{Ping,\min}^{[P_X \circ P_Y]} \leq Round\_Trip\_Delay_{meas}^{[P_X \circ P_Y]} \leq Round\_Trip\_Delay_{Ping,\max}^{[P_X \circ P_Y]} \tag{112}$$

[0086] Using the Round_Trip_Delay properties and the Ping_Time relationships, the maximum Round_Trip_Delay between two given leaf nodes can be bounded for any possible topology.

[0087] The simplest and most accurate Round_Trip_Delay determination is afforded when the Bus Manager is one of the leaf nodes in question as shown in FIG. 9.

From (92),

$$Round\_Trip\_Delay_{\max}^{[P_{BM} \circ P_Y]} \leq Round\_Trip\_Delay_{meas_i}^{[P_{BM} \circ P_Y]} + \sum_{n}^{(BM,Y)} 2 \cdot Jitter_n \quad (113)$$

And from (112),

$$Round\_Trip\_Delay_{\max}^{[P_{BM} \circ P_Y]} \leq Round\_Trip\_Delay_{Ping,\max}^{[P_{BM} \circ P_Y]} + \sum_{n}^{(BM,Y)} 2 \cdot Jitter_n \quad (114)$$

Likewise, the reverse path is also bounded:

$$Round\_Trip\_Delay_{\max}^{[P_Y \circ P_{BM}]} \leq Round\_Trip\_Delay_{Ping,\max}^{[P_{BM} \circ P_Y]} + \sum_{n}^{(BM,Y)} 2 \cdot Jitter_n \quad (115)$$

[0088] The second topology to consider is when the bus manager is not a leaf but is part of the connecting path between the two leaves as illustrated in FIG. 10.

[0089] Expressing the max delay piecewise,

$$\begin{aligned} Round\_Trip\_Delay_{\max}^{[P_X \circ P_Y]} = {} & Round\_Trip\_Delay_{\max}^{[P_X \circ P_{BM}]} + Round\_Trip\_Delay_{\max}^{[P_{BM} \circ P_Y]} + \\ & PHY\_DELAY_{BM,\max}^{P'_{BM} \to P_{BM}} + ARB\_RESPONSE\_DELAY_{BM,\max}^{P_{BM} \to P'_{BM}} \end{aligned} \quad (116)$$

[0090] Equations (92) and (96) allow:

$$
\begin{aligned}
Round\_Trip\_Delay_{\max}^{[P_X \circ P_Y]} \le\ & Round\_Trip\_Delay_{meas}^{[P_{BM} \circ P_X]} + \sum_{n}^{(BM,X)} 2 \cdot Jitter_n + \\
& Round\_Trip\_Delay_{meas}^{[P_{BM} \circ P_Y]} + \sum_{n}^{(BM,Y)} 2 \cdot Jitter_n + \\
& PHY\_DELAY_{BM,\max}^{P'_{BM} \rightarrow P_{BM}} + ARB\_RESPONSE\_DELAY_{BM,\max}^{P_{BM} \rightarrow P'_{BM}}
\end{aligned}
\tag{117}
$$

And from (112),

$$
\begin{aligned}
Round\_Trip\_Delay_{\max}^{[P_X \circ P_Y]} \le\ & Round\_Trip\_Delay_{Ping,\max}^{[P_{BM} \circ P_X]} + \sum_{n}^{(BM,X)} 2 \cdot Jitter_n + \\
& Round\_Trip\_Delay_{Ping,\max}^{[P_{BM} \circ P_Y]} + \sum_{n}^{(BM,Y)} 2 \cdot Jitter_n + \\
& PHY\_DELAY_{BM,\max}^{P'_{BM} \rightarrow P_{BM}} + ARB\_RESPONSE\_DELAY_{BM,\max}^{P_{BM} \rightarrow P'_{BM}}
\end{aligned}
\tag{118}
$$

[0091] Likewise, the reverse path is also bounded:

$$
\begin{aligned}
Round\_Trip\_Delay_{\max}^{[P_Y \circ P_X]} \le\ & Round\_Trip\_Delay_{Ping,\max}^{[P_{BM} \circ P_X]} + \sum_{n}^{(BM,X)} 2 \cdot Jitter_n + \\
& Round\_Trip\_Delay_{Ping,\max}^{[P_{BM} \circ P_Y]} + \sum_{n}^{(BM,Y)} 2 \cdot Jitter_n + \\
& PHY\_DELAY_{BM,\max}^{P_{BM} \rightarrow P'_{BM}} + ARB\_RESPONSE\_DELAY_{BM,\max}^{P'_{BM} \rightarrow P_{BM}}
\end{aligned}
\tag{119}
$$

[0092] The final topology to consider is when the bus manager is not a leaf but is not part of the connecting path between the two leaves as illustrated in FIG. 11.

[0093] Expressing the max delay piecewise,

$$
\begin{aligned}
Round\_Trip\_Delay_{\max}^{[P_X \circ P_Y]} = {} & Round\_Trip\_Delay_{\max}^{[P_X \circ P_N]} + Round\_Trip\_Delay_{\max}^{[P_N \circ P_Y]} + \\
& PHY\_DELAY_{N,\max}^{P_N' \to P_N} + ARB\_RESPONSE\_DELAY_{N,\max}^{P_N \to P_N'}
\end{aligned}
\tag{120}
$$

[0094] Equations (107) and (101) allow:

$$
\begin{aligned}
Round\_Trip\_Delay_{\max}^{[P_X \circ P_Y]} \le {} & \left(\begin{array}{l}
Round\_Trip\_Delay_{meas_1}^{[P_{BM} \circ P_X]} + \sum_{n}^{(BM,X)} 2 \cdot Jitter_n - \\
Round\_Trip\_Delay_{meas_2}^{[P_{BM} \circ P_N]} - PHY\_DELAY_{N,\min}^{P_N^{BM} \to P_N'} - \\
ARB\_RESPONSE\_DELAY_{N,\min}^{P_N' \to P_N^{BM}} - 2 \cdot Jitter_N
\end{array} \right) + \\
& \left(\begin{array}{l}
Round\_Trip\_Delay_{meas_1}^{[P_{BM} \circ P_Y]} + \sum_{n}^{(BM,Y)} 2 \cdot Jitter_n - \\
Round\_Trip\_Delay_{meas_2}^{[P_{BM} \circ P_N]} - PHY\_DELAY_{N,\min}^{P_N^{BM} \to P_N} - \\
ARB\_RESPONSE\_DELAY_{N,\min}^{P_N \to P_N^{BM}} - 2 \cdot Jitter_N
\end{array} \right) + \\
& PHY\_DELAY_{N,\max}^{P_N' \to P_N} + ARB\_RESPONSE\_DELAY_{N,\max}^{P_N \to P_N'}
\end{aligned}
\tag{121}
$$

[0095] And from (112),

$$Round\_Trip\_Delay_{\max}^{[P_X \circ P_Y]} \leq \begin{pmatrix} Round\_Trip\_Delay_{Ping,\max}^{[P_{BM} \circ P_X]} + \sum_{n}^{(BM,X)} 2 \cdot Jitter_n + \\ Round\_Trip\_Delay_{Ping,\max}^{[P_{BM} \circ P_Y]} + \sum_{n}^{(BM,Y)} 2 \cdot Jitter_n + \\ PHY\_DELAY_{N,\max}^{P_N^{'} \to P_N} + ARB\_RESPONSE\_DELAY_{N,\max}^{P_N \to P_N^{'}} \end{pmatrix} - \begin{pmatrix} 2 \cdot Round\_Trip\_Delay_{Ping,\min}^{[P_{BM} \circ P_N]} + 4 \cdot Jitter_N + \\ PHY\_DELAY_{N,\min}^{P_N^{BM} \to P_N^{'}} + ARB\_RESPONSE\_DELAY_{N,\min}^{P_N^{'} \to P_N^{BM}} + \\ PHY\_DELAY_{N,\min}^{P_N^{BM} \to P_N} + ARB\_RESPONSE\_DELAY_{N,\min}^{P_N \to P_N^{BM}} \end{pmatrix} \quad (122)$$

ARB_RESPONSE_DELAY is a difficult parameter to characterize. Proper PHY operation requires that arb signals propagate at least as fast as the data bits, otherwise the arbitration indications could shorten as they are repeated through a network. This fact places a bound on the maximum ARB_RESPONSE_DELAY: ARB_RESPONSE_DELAY between two ports at a particular instant must always be less than or equal to the data repeat delay at the very same instant. Although the distinction is subtle, this is not the same as saying the maximum ARB_RESPONSE_DELAY is PHY_DELAY. (PHY_DELAY only applies to the first bit of a packet and is known to have some jitter from one repeat operation to the next. Consequently, requiring ARB_RESPONSE_DELAY <= PHY_DELAY doesn't force ARB_RESPONSE_DELAY to track the instantaneous PHY_DELAY nor does it allow ARB_RESPONSE_DELAY to track the data repeat time for the last bit of a packet which may actually exceed PHY_DELAY due to PPM drift.) Finally, the table approach to calculating gap_counta and gap_countb rely on ARB_RESPONSE_DELAY always being bounded by the maximum PHY_DELAY when determining the Round_Trip_Delay.

[0096] The minimum ARB_RESPONSE_DELAY is only of significance when calculating Data_Arb_Mismatch as required by gap_countc and gap_countd. Ideally, Data_Arb_Mismatch should be a constant regardless of PHY_DELAY so that neither gap_countc nor gap_countd will begin to dominate the gap_count setting as PHY_DELAY increases. Consequently, the minimum ARB_RESPONSE_DELAY should track the instantaneous PHY_DELAY with some offset for margin. Simply specifying the min value as a function of PHY_DELAY is ambiguous, however, since PHY_DELAY can be easily confused with the max DELAY reported in the register map. (For example, with DELAY at 144 ns, it would be easy to assume a min of PHY_DELAY – 60 ns would be equivalent to 84 ns. But if the worst case first bit repeat delay was only 100 ns, arb signals repeating with a delay of 40 ns ought to be considered within spec even though the delay is < 84 ns.)

[0097] Consequently, specifying an upper and a lower bound for ARB_RESPONSE_DELAY is best done in the standard with words rather than values. The minimum and maximum values for ARB_RESPONSE_DELAY include that between all ordered pairs of ports, the PHY shall repeat arbitration line states at least as fast as clocked data, but not more than 60 ns faster than clocked data.

[0098] A better approach is to replace ARB_RESPONSE_DELAY with the parameter DELAY_MISMATCH which is defined in the comment column as "Between all ordered pairs of ports, the instantaneous repeat delay for data less the instantaneous repeat delay for arbitration line states." Then, the minimum would be given as 0 ns and the maximum would be 60 ns.

[0099] For a table based calculation of Round_Trip_Delay, either approach above allows the use of PHY_DELAY(max) for ARB_RESPONSE_DELAY. Since Round_Trip_Delay considers the arbitration repeat delay in the direction opposite to the original packet flow, the return arbitration indication of interest is known to arrive at the receive port when the PHY is idle (all caught up with nothing to repeat). At that point, the instantaneous PHY_DELAY is the same as the first data bit repeat delay which is bounded by PHY_DELAY(max). Since ARB_RESPONSE_DELAY is always bounded by the instantaneous PHY_DELAY, it to is bounded by PHY_DELAY(max) at the point the arbitration indication first arrives.

[0100] The minimum bound on PHY_DELAY is used by the bus manager when determining the round_trip_delay between leaf nodes that are not separated by the bus manager. The more precise the minimum bound, the more accurate the pinging calculation can be. Ideally then, the bound may want to scale with increasing PHY_DELAY. Alternatively, the lower bound could be calculated by examining the Delay field in the register map: if zero, the lower bound is assumed to be the fixed value specified (60 ns currently). If non-zero, the lower bound could then be determined by subtracting the jitter field (converted to ns) from the delay field (converted to ns).

[0101] The "Jitter" field was introduced to aid in selection of gap_count via pinging by describing the uncertainty found in any empirical measurement of Round_Trip_Delay. Since Round_Trip_Delay encompasses an "outbound" PHY_DELAY and a "return" ARB_RESPONSE_DELAY, the jitter term should capture uncertainty in both. The needs of pinging can be met with the following

description for jitter: Upper bound of the mean average of the worst case data repeat jitter (max/min variance) and the worst case arbitration repeat jitter (max/min variance), expressed as 2*(jitter + 1)/BASE_RATE.

[0102] Note that from the discussion on minimum PHY_DELAY, it may be desirable to require that if the delay field is non-zero, then the slowest first data bit repeat delay can be calculated by subtracting the jitter value from the delay value.